

# Visual Domain Adaptation: A Survey of Recent Advances

Vishal M. Patel, *Member, IEEE*, Raghuraman Gopalan, *Member, IEEE*, Ruonan Li, and  
Rama Chellappa, *Fellow, IEEE*

## Abstract

In pattern recognition and computer vision, one is often faced with scenarios where the training data used to learn a model has different distribution from the data on which the model is applied. Regardless of the cause, any distributional change that occurs after learning a classifier can degrade its performance at test time. Domain adaptation tries to mitigate this degradation. In this paper, we provide a survey of domain adaptation methods for visual recognition. We discuss the merits and drawbacks of existing domain adaptation approaches and identify promising avenues for research in this rapidly evolving field.

## Index Terms

Computer vision, object recognition, domain adaptation, scene understanding.

## I. INTRODUCTION

Supervised learning techniques have made tremendous contributions to machine learning and computer vision, leading to the development of robust algorithms that are applicable in practical scenarios. While

Vishal M. Patel is with the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: pvishalm@umd.edu).

Raghuraman Gopalan is with the Department of Video and Multimedia Technologies Research, AT&T Labs-Research, Middletown NJ 07748 (e-mail: raghuram@research.att.com).

Ruonan Li, is with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138 (e-mail: ruonanli@seas.harvard.edu).

Rama Chellappa is with the Department of Electrical and Computer Engineering and the Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742 (e-mail: rama@umiacs.umd.edu).

EDICS: SMR-REP, ARS-SRE

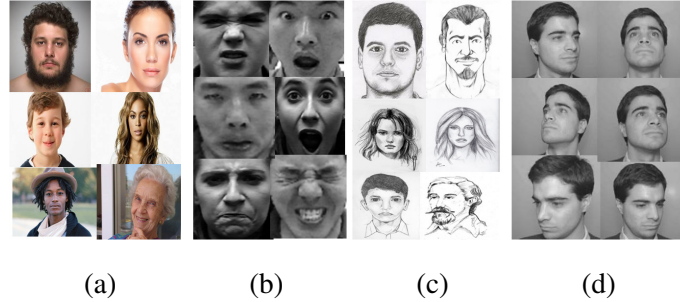


Fig. 1: (a) Unconstrained face images. (b) Images with expression variations. (c) Sketch images. (d) Images with pose variations. Real-world object recognition algorithms, such as face recognition, must learn to adapt to distributions specific to each domain shown in (a)-(d).

these algorithms have advanced the state-of-the-art significantly, their performance is often limited by the amount of labeled training data available. Labeling is expensive and time consuming due to the significant amount of human efforts involved. However, collecting unlabeled visual data is becoming considerably easier due to the availability of low cost consumer and surveillance cameras, and large internet databases such as Flickr and YouTube. These data often come from multiple sources and modalities. Thus, when designing a classification or retrieval algorithm using these heterogeneous data, one has to constantly deal with the changing distribution of these data samples. Examples of such cases include: recognizing objects under poor lighting conditions and poses while algorithms are trained on well-illuminated objects at frontal pose, detecting and segmenting an organ of interest from MRI images when available algorithms are instead optimized for CT and X-Ray images, recognizing and detecting human faces on infrared images while algorithms are optimized for color images, etc.

This challenge is commonly referred to as *covariate shift* [1] or *data set bias* [2], [3]. Any distributional change or domain shift that occurs after training can degrade the performance at test time. For instance, in the case of face recognition, to achieve useful performance *in the wild*, face representation and recognition methods must learn to adapt to distributions specific to each application domain shown in Fig. 1 (a)-(d). Domain adaptation tackles this problem by leveraging domain shift characteristics from labeled data in a related domain when learning a classifier for unseen data. Although some special kinds of domain adaptation problems have been studied under different names such as covariate shift [1], class imbalance [4], and sample selection bias [5], [6], it only started gaining significant interest very recently in computer vision. There are also some closely related but not equivalent machine learning problems that

have been studied extensively, including transfer learning or multi-task learning [7], self-taught learning [8], semi-supervised learning [9] and multiview analysis [10]. A review of domain adaptation methods from machine learning and natural language processing communities can be found in [11]. Our goal in this paper is to survey recent domain adaptation approaches for computer vision applications, discuss their advantages and disadvantages, and identify interesting open problems.

Rest of the paper is organized as follows. The domain adaptation learning problem is formulated in Section II. Various visual domain adaptation methods are reviewed in Section III. Applications of domain adaptation in object and face recognition are presented in Section IV. Finally, concluding remarks are made in Section V.

## II. NOTATION AND RELATED LEARNING PROBLEMS

In this section, we introduce the notation and formulate the domain adaptation learning problem. Furthermore, we discuss similarities and differences among the various learning problems related to domain adaptation.

### A. Notation and Formulation

We refer to the training dataset with plenty of labeled data as the source domain and the test dataset with a few labeled data or no labeled data as the target domain. Following [11], let  $X$  and  $Y$  denote the input (data) and the output (label) random variables, respectively. Let  $P(X, Y)$  denote the joint probability distribution of  $X$  and  $Y$ . In domain adaptation, the target distribution is generally different than the source distribution and the true underlying joint distribution  $P(X, Y)$  is unknown. We have two different distributions one for the target domain and the other for the source domain. We denote the joint distribution in the source domain and the target domain as  $P_s(X, Y)$  and  $P_t(X, Y)$ , respectively. The marginal distributions of  $X$  and  $Y$  in the source and the target domains are denoted by  $P_s(X), P_s(Y), P_t(X), P_t(Y)$ , respectively. Similarly, the conditional distributions in the two domains are denoted by  $P_s(X|Y), P_s(Y|X), P_t(X|Y), P_t(Y|X)$ . The joint probability of  $X = x$  and  $Y = y$  is denoted by  $P(X = x, Y = y) = P(x, y)$ . Here,  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$ , where  $\mathcal{X}$  and  $\mathcal{Y}$  denote the instance space and class label spaces, respectively.

Let  $\mathcal{S} = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ , where  $\mathbf{x}^s \in \mathbb{R}^N$  denote the labeled data from the source domain. Here,  $\mathbf{x}^s$  is referred to as an observation and  $y^s$  is the corresponding class label. Labeled data from the target domain is denoted by  $\mathcal{T}_l = \{(\mathbf{x}_i^{tl}, y_i^{tl})\}_{i=1}^{N_{tl}}$  where  $\mathbf{x}^{tl} \in \mathbb{R}^M$ . Similarly, unlabeled data in the target domain is denoted by  $\mathcal{T}_u = \{\mathbf{x}_i^{tu}\}_{i=1}^{N_{tu}}$  where  $\mathbf{x}^{tu} \in \mathbb{R}^M$ . Unless specified otherwise, we assume  $N = M$ . Let

$\mathcal{T} = \mathcal{T}_l \cup \mathcal{T}_u$ . As a result, the total number of samples in the target domain is denoted by  $N_t$  which is equal to  $N_{tl} + N_{tu}$ . Denote  $\mathbf{S} = [\mathbf{x}_1^s, \dots, \mathbf{x}_{N_s}^s]$  as the matrix of  $N_s$  data points from  $\mathcal{S}$ . Denote  $\mathbf{T}_l = [\mathbf{x}_1^{tl}, \dots, \mathbf{x}_{N_{tl}}^{tl}]$  as the matrix of  $N_{tl}$  data from  $\mathcal{T}_l$ ,  $\mathbf{T}_u = [\mathbf{x}_1^{tu}, \dots, \mathbf{x}_{N_{tu}}^{tu}]$  as the matrix of  $N_{tu}$  data from  $\mathcal{T}_u$  and  $\mathbf{T} = [\mathbf{T}_l | \mathbf{T}_u] = [\mathbf{x}_1^t, \dots, \mathbf{x}_{N_t}^t]$  as the matrix of  $N_t$  data from  $\mathcal{T}$ .

It is assumed that both the target and source data pertain to  $C$  classes or categories. Furthermore, it is assumed that all categories have some labeled data. We assume there is always a relatively large amount of labeled data in the source domain and a small amount of labeled data in the target domain. As a result,  $N_s \gg N_{tl}$ .

The goal of domain adaptation is to learn a function  $f(\cdot)$  that predicts the class label of a novel test sample from the target domain. Depending on the availability of the source and target domain data, the domain adaptation problem can be defined in many different ways:

- In *semi-supervised domain adaptation*, the function  $f(\cdot)$  is learned using the knowledge in  $\mathcal{S}$  and  $\mathcal{T}_l$ .
- In *unsupervised domain adaptation*, the function  $f(\cdot)$  is learned using the knowledge in  $\mathcal{S}$  and  $\mathcal{T}_u$ .
- In *multi-source domain adaptation*,  $f(\cdot)$  is learned from more than one domain in  $\mathcal{S}$  accompanying each of the above two cases.
- Finally, in the *heterogeneous domain adaptation*, the dimensions of features in the source and target domains are assumed to be different. In other words,  $N \neq M$ .

## B. Related Approaches

1) *Covariate Shift*: One variation of the domain adaptation problem is where given an observation, the conditional distributions of  $Y$  are the same in the source and the target domains but the marginal distributions of  $X$  differ in the two domains. In other words,  $P_t(Y|X = x) = P_s(Y|X = x)$  for all  $x \in \mathcal{X}$ , but  $P_t(X) \neq P_s(X)$ . This resulting difference between the two domains is known as covariate shift [1] or sample selection bias [5], [6].

Instance weighting methods can be used to address this covariate shift problem in which estimated weights are incorporated into a loss function in an attempt to make the weighted training distribution look like the testing distribution [11]. To see this, let us briefly review the empirical risk minimization framework for supervised learning [12]. Let  $\theta \in \Theta$  be a model family from which we want to select an optimal parameter  $\theta^*$  for the inference. Let  $g(x, y, \theta)$  be a loss function. We want to minimize the

following objective function

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} P(x,y) g(x,y,\theta)$$

to obtain the optimal  $\theta^*$  for the distribution  $P(X,Y)$ . Since  $P(X,Y)$  is unknown, we use the empirical distribution  $\tilde{P}(X,Y)$  to estimate  $P(X,Y)$ . A good model  $\hat{\theta}$  can be found by minimizing the following empirical risk

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \tilde{P}(x,y) g(x,y,\theta) \\ &= \arg \min_{\theta \in \Theta} \sum_{i=1}^N g(x_i, y_i, \theta), \end{aligned}$$

where  $\{(x_i, y_i)\}_{i=1}^N$  is a set of training instances randomly sampled from  $P(X,Y)$ . This formulation can be extended to domain adaptation by minimizing the following expected loss over the target domain distribution to find the optimal model parameter for the target domain [11]

$$\theta_t^* = \arg \min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} P_t(x,y) g(x,y,\theta).$$

In domain adaptation setting, the training instances  $\{(x_i^s, y_i^s)\}_{i=1}^{N_s}$  are randomly sampled from the source distribution  $P_s(X,Y)$ . As a result, we get

$$\begin{aligned} \theta_t^* &= \arg \min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \frac{P_t(x,y)}{P_s(x,y)} P_s(x,y) g(x,y,\theta) \\ &\approx \arg \min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} \frac{P_t(x,y)}{P_s(x,y)} \tilde{P}_s(x,y) g(x,y,\theta) \\ &= \arg \min_{\theta \in \Theta} \sum_{i=1}^{N_s} \frac{P_t(x_i^s, y_i^s)}{P_s(x_i^s, y_i^s)} g(x_i^s, y_i^s, \theta). \end{aligned} \tag{1}$$

As can be seen from (1), weighting the loss of the source samples by  $\frac{P_t(x,y)}{P_s(x,y)}$  provides a solution to the domain adaptation problem [11].

Under covariate shift, the ratio  $\frac{P_t(x,y)}{P_s(x,y)}$  can be rewritten as follows

$$\begin{aligned} \frac{P_t(x,y)}{P_s(x,y)} &= \frac{P_t(x) P_t(y|x)}{P_s(x) P_s(y|x)} \\ &= \frac{P_t(x)}{P_s(x)}. \end{aligned}$$

As a result, one can weigh each training instance with  $\frac{P_t(x)}{P_s(x)}$ . Shimodaira [1] explored this approach to reweight the log likelihood of each training instance using  $\frac{P_t(x)}{P_s(x)}$  for covariate shift. Various methods can

be used to estimate the ratio  $\frac{P_t(x)}{P_s(x)}$ . For instance, non-parametric density estimation [1], [13] and Kernel Mean Match-based methods [14] have been proposed in the literature to directly estimate the ratio.

2) *Class Imbalance*: Another special case of the domain adaptation formulation assumes that  $P_t(X|Y = y) = P_s(X|Y = y)$  for all  $y \in \mathcal{Y}$ , but  $P_t(Y) \neq P_s(Y)$ . This difference is often known as class imbalance [4]. Under this assumption, the ratio in (1) can be rewritten as follows

$$\begin{aligned} \frac{P_t(x, y)}{P_s(x, y)} &= \frac{P_t(y)}{P_s(y)} \frac{P_t(x|y)}{P_s(x|y)} \\ &= \frac{P_t(y)}{P_s(y)}. \end{aligned}$$

As a result, one only needs to consider  $\frac{P_t(y)}{P_s(y)}$  to weigh the instances [15].

Re-sampling can also be applied on the training instances from the source domain so that the re-sampled data roughly has the same class distribution as the target domain. In these methods, under-represented classes are over-sampled and over-represented classes are under-sampled [11].

3) *Transfer Learning*: Multitask learning or transfer learning is closely related to domain adaptation [7], [16]. In multitask learning, different tasks are considered but the marginal distribution of the source and target data are similar. In other words, assuming  $L$  tasks, the joint probability of each task  $\{P(X, Y_i)\}_{i=1}^L$  are different but there is only a single distribution  $P(X)$  of the observation. When learning the class conditional models  $\{P(Y_i|X, \theta_i)\}_{i=1}^L$  for  $L$  tasks, it is assumed that the model parameters of the individual tasks are drawn from a common prior distribution  $P_\Theta(\theta)$ .

Since domain adaptation considers only a single task but different domains, it is a somewhat different problem than multitask learning. However, one can view domain adaptation as a special case of multitask learning with two tasks, one on the source domain and the other on the target domain. In fact, some domain adaptation methods are essentially solving transfer learning problems. We refer the readers to [16] for a comprehensive survey on various transfer learning methods.

4) *Semi-supervised Learning*: The performance of a supervised classification algorithm is often dependent on the availability of sufficient amount of training data. However, labeling samples is expensive and time consuming due to the significant human effort involved. As a result, it is desirable to have methods that learn a classifier with high accuracy from only a limited number of labeled training data. In semi-supervised learning, unlabeled data is exploited to remedy the lack of labeled data. This in turn requires that the unlabeled data comes from the same distribution as the labeled data. Hence, if we ignore the domain difference, and treat the labeled source instances as labeled data and the unlabeled target domain instances as unlabeled data, then the resulting problem is that of the semi-supervised learning problem. As a result, one can apply any semi-supervised learning algorithm [9] to the domain adaptation problem.

The subtle difference between domain adaptation and semi-supervised learning comes from the following two facts [11]

- The amount of labeled data in semi-supervised learning is small but large in domain adaptation, and
- The labeled data may be noisy in domain adaptation if one does not assume  $P_s(Y|X = x) = P_t(Y|X = x)$  for all  $x$ , whereas in semi-supervised learning the labeled data is assumed to be reliable.

In fact, there have been several works in the literature that extend semi-supervised learning methods to domain adaptation. A naive Bayes transfer classifier algorithm which allows for the training and test data distributions to be different for text classification was proposed in [17]. This algorithm first estimates the initial probabilities under a distribution of one labeled data set, and then uses an Expectation Maximization (EM) algorithm to revise the model for a different distribution of the test data which are assumed to be unlabeled. This EM-based domain adaptation method can be shown to be equivalent to a semi-supervised EM algorithm [18]. Some of the other methods that extend domain adaptation using semi-supervised learning include [19], [20].

5) *Self-taught Learning*: Another problem related to domain adaptation and semi-supervised learning is self-taught learning [8], [21]. In self-taught learning, we are given limited data for a classification task and also large amounts of unlabeled data that is only mildly related to the task. In particular, the unlabeled data may not arise from the same distribution or share the class labels. This assumption essentially differentiates self-taught learning from semi-supervised learning. Self-taught learning is motivated by the observation that many randomly downloaded images contain basic visual features such as edges and corners that are similar to those in the training images. As a result, if one is able to learn to recognize such patterns from the unlabeled data, then these features can be used for the supervised learning task of interest [8].

A sparse coding-based approach was proposed in [8] for self-taught learning, where a dictionary is learned using unlabeled data. Then, higher level features are computed by solving a convex  $\ell_1$ -regularized least squares problem using the learned dictionary and the labeled training data. Finally, a classifier is trained by applying a supervised learning algorithm such as an SVM on these higher level labeled features. A discriminative version of this algorithm was also presented in [22]. Furthermore, an unsupervised self-taught learning algorithm called self-taught clustering was proposed in [23]. Self-taught clustering aims at clustering small collection of target unlabeled data with the help of a large amount of auxiliary unlabeled data. It is assumed that the target and auxiliary data have different distribution. It was shown that this algorithm can greatly outperform several state-of-the-art clustering methods when utilizing irrelevant

unlabeled data.

6) *Multiview Analysis*: In many computer vision applications, data often comes in multiple views or styles. For instance, in object recognition, one has to deal with objects in different poses (views) and lighting conditions. As a result, one is faced with the problem of classifying or retrieving objects where the source (gallery) and target (query) data belong to different views. A direct comparison of instances across different views is not meaningful since they lie in different feature spaces.

In a multiview (also known as cross view or multi-modal) learning setting, correspondences are assumed to be known between the two view samples. In other words, samples are often given in pairs corresponding to different views. This assumption essentially differentiates cross view learning from domain adaptation, where no correspondences are assumed between the domain samples. One popular solution in multiview learning is to learn view-specific projection directions using the paired samples from different views (domains) into a common latent space [10]. Classification or retrieval can then be performed in the latent space where both the target and source data share the same feature space. Other methods for multiview learning include [24], [25], [26], [27], [28].

### III. VISUAL DOMAIN ADAPTATION APPROACHES

Domain adaptation is a fundamental problem in machine learning and has gained a lot of traction in natural language processing, statistics, machine learning and recently in computer vision. Early visual domain adaptation methods were applied to domain shift in videos [29], [30]. In particular, Duan *et al.* [30] proposed to adapt video concept classifiers between news videos collected from different news channels. Since then, there have been a plethora of approaches proposed in the vision literature for object category adaptation. In what follows, we present a number of recent domain adaptation strategies for visual recognition.

#### A. Feature Augmentation-based Approaches

One of the simplest domain adaptation approaches is the feature augmentation work of DauméIII [31]. The goal is to make a domain specific copy of the original features for each domain. Each feature in the original domain of dimension  $N$  is mapped onto an augmented space of dimension  $3N$  simply by duplicating the feature vectors. The augmented feature maps for the source and target domains are defined as

$$\Phi^s(\mathbf{x}_i^s) = \begin{bmatrix} \mathbf{x}_i^s \\ \mathbf{x}_i^s \\ \mathbf{0}_N \end{bmatrix}, \quad \Phi^t(\mathbf{x}_i^{tl}) = \begin{bmatrix} \mathbf{x}_i^{tl} \\ \mathbf{0}_N \\ \mathbf{x}_i^{tl} \end{bmatrix} \quad (2)$$



where  $\mathbf{x}_i^s \in \mathcal{S}$ ,  $\mathbf{x}_i^{tl} \in \mathcal{T}_l$ , and  $\mathbf{0}_N$  denotes a zero vector of dimension  $N$ . The first  $N$ -dimensional component of this augmented feature corresponds to commonality between source and target, the second  $N$ -dimensional component corresponds to the source while the last component corresponds to the target domain. Both source and target domain features are transformed using these augmented feature map and the resulting feature is passed onto the underlying supervised classifier. It was shown in [31] that when linear classifiers are used, this feature augmentation method is equivalent to decomposing the model parameter  $\theta_i$  for domain  $i$  into  $\tilde{\theta}_i + \theta_c$ , where  $\theta_c$  is shared by all domains. This ‘‘frustratingly easy’’ feature augmentation framework can be easily extended to a multi-domain case by making more copies of the original feature space. Furthermore, a kernel version of this method is also derived in [31].

A feature augmentation-based method for utilizing the heterogeneous data from the source and target domains was recently proposed in [32]. The approach taken in [32] is to introduce a common subspace for the source and target data so that the heterogeneous features from two domains can be compared. In particular, both the source and target data of dimension  $N$  and  $M$ , respectively are projected onto a latent domain of dimension  $l$  using two projection matrices  $\mathbf{W}_1 \in \mathbb{R}^{l \times N}$  and  $\mathbf{W}_2 \in \mathbb{R}^{l \times M}$ , respectively. The augmented feature maps for the source and target domains in the common space are then defined as

$$\Phi^s(\mathbf{x}_i^s) = \begin{bmatrix} \mathbf{W}_1 \mathbf{x}_i^s \\ \mathbf{x}_i^s \\ \mathbf{0}_M \end{bmatrix} \in \mathbb{R}^{l+N+M}, \quad (3)$$

$$\Phi^t(\mathbf{x}_i^{tl}) = \begin{bmatrix} \mathbf{W}_2 \mathbf{x}_i^{tl} \\ \mathbf{0}_N \\ \mathbf{x}_i^{tl} \end{bmatrix} \in \mathbb{R}^{l+N+M}, \quad (4)$$

where  $\mathbf{x}_i^s \in \mathcal{S}$ ,  $\mathbf{x}_i^{tl} \in \mathcal{T}_l$ , and  $\mathbf{0}_M$  is an  $M$ -dimensional zero vector. Once, the data from both domains are transformed onto a common space, they can be readily passed onto a supervised classifier [32]. Fig. 2 illustrates an overview of this method.

The general idea behind ‘‘frustratingly easy’’ feature augmentation method of DauméIII [31] has been extended to consider a manifold of intermediate domains [33], [34]. Manifold-based methods for unsupervised visual domain adaptation were first proposed by Gopalan *et al.* in [33]. Rather than working with the information conveyed by the source and target domains alone, [33] proposes to use incremental learning by gradually following the geodesic path between the source and target domains. Geodesic flows are used to derive intermediate subspaces that interpolate between the source and target domains. Fig. 3 shown an overview of this method.

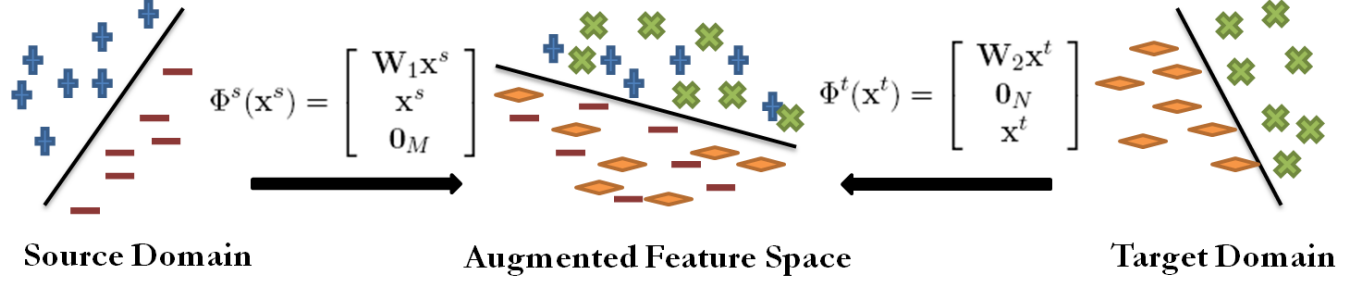


Fig. 2: By using two projection matrices  $\mathbf{W}_1$  and  $\mathbf{W}_2$ , one can transform the heterogeneous samples from two domains into an augmented feature space [32].

It is assumed that the dimension of features in both the source and target domains is the same, e.g.  $N = M$ . First, PCA is applied on  $\mathbf{S}$  and  $\mathbf{T}_u$ , which generates two  $l$ -dimensional subspaces denoted by two matrices  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , respectively, where  $l < N$ . The space of  $l$ -dimensional subspaces in  $\mathbb{R}^N$  containing origin can be identified with the Grassmann manifold  $\mathbb{G}_{N,l}$ . As a result,  $\mathbf{S}_1$  and  $\mathbf{S}_2$  can be viewed as points on  $\mathbb{G}_{N,l}$ . By viewing  $\mathbb{G}_{N,l}$  as quotient space of  $SO(N)$ ,<sup>1</sup> the geodesic path in  $\mathbb{G}_{N,l}$  starting from  $\mathbf{S}_1$  is given by a one-parameter exponential flow  $\Psi(t') = \mathbf{Q} \exp(t' \mathbf{B}) \mathbf{J}$ , where  $\exp$  refers to the matrix exponential,  $\mathbf{Q} \in SO(N)$  such that  $\mathbf{Q}^T \mathbf{S}_1 = \mathbf{J}$  and  $\mathbf{J} = \begin{bmatrix} \mathbf{I}_l \\ \mathbf{0}_{N-l,l} \end{bmatrix}$ . Here,  $\mathbf{I}_l$  is a  $l \times l$  identity matrix and  $\mathbf{B}$  is a skew-symmetric, block-diagonal matrix of the form  $\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{bmatrix}$ ,  $\mathbf{A} \in \mathbb{R}^{(N-l) \times l}$ , where  $(\cdot)^T$  denotes the transposition operation and the sub-matrix  $\mathbf{A}$  specifies the direction and the speed of geodesic flow. The geodesic flow between  $\mathbf{S}_1$  and  $\mathbf{S}_2$  is obtained by computing the direction matrix  $\mathbf{A}$  such that the geodesic along that direction, while starting from  $\mathbf{S}_1$ , reaches  $\mathbf{S}_2$  in unit time. The matrix  $\mathbf{A}$  is computed using the inverse exponential mapping. Once  $\mathbf{A}$  is computed, the expression for  $\Psi(t')$  is used to obtain the intermediate subspaces between  $\mathbf{S}_1$  and  $\mathbf{S}_2$  by varying the value of  $t'$  between 0 and 1.

Let  $\mathbf{S}'$  be the collection of subspaces  $\mathbf{S}_t, t \in \mathbb{R}, 1 \leq t \leq 2$ , which includes  $\mathbf{S}_1$  and  $\mathbf{S}_2$  and all intermediate subspaces. Let  $k$  denote the total number of such subspaces. The intermediate cross-domain data representations  $\mathbf{U}$  is obtained by projecting the source data  $\mathbf{S}$  and the target data  $\mathbf{T}_u$  onto  $\mathbf{S}'$ . The

<sup>1</sup>Here,  $SO(N)$  represents the special orthogonal group which is the group of orthogonal  $N \times N$  matrices with determinant 1.

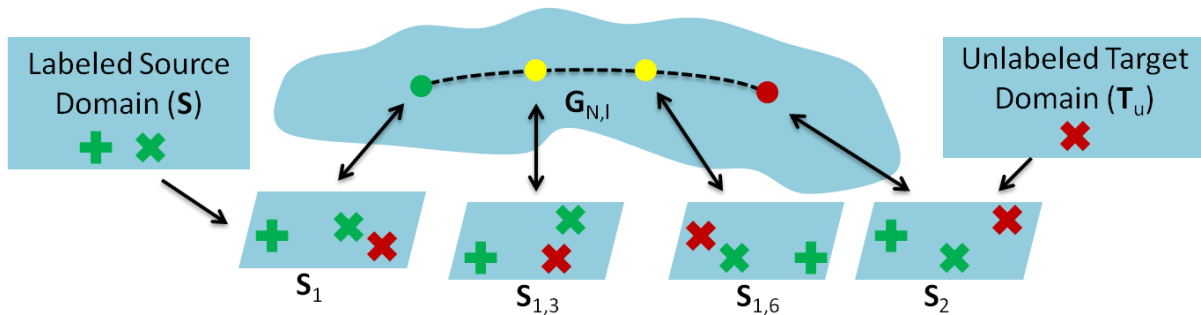


Fig. 3: An overview of the manifold-based unsupervised domain adaptation method [33]. With labeled data  $\mathbf{S}$  from source domain corresponding to two classes  $+$  and  $\times$ , and unlabeled data  $\mathbf{T}_u$  from target domain belonging to class  $\times$ , generative subspaces  $\mathbf{S}_1$  and  $\mathbf{S}_2$  are derived using PCA. Then, by viewing  $\mathbf{S}_1$  and  $\mathbf{S}_2$  as points on the Grassmann manifold  $\mathbb{G}_{N,l}$  (green and red dots), points along the geodesic between them (dashed line) are sampled to obtain geometrically meaningful intermediate subspaces (yellow dots).

final feature representation of dimension  $lk$  is obtained by projecting data onto  $k$  different subspaces. A model on these extended features is learned using PLS and the assignment of target labels is performed using the nearest neighbor method [33]. A non-linear version of this method, as well as extension to semi-supervised domain adaptation have also been presented in [34]. Furthermore, assuming that the domain to which samples belong to has been identified a priori [35], [36] this method has been extended to multi-domain adaptation in [34].

Recently, the approach of [33] was kernelized and extended to the infinite case, defining a new kernel equivalent to integrating over all common subspaces that lie on the geodesic flow connecting the source and target subspaces  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , respectively [37], [38], [39]. Furthermore, assuming that the data lies in a union of subspaces in both source and target domains, a framework based on the parallel transport of union of the source subspaces on the Grassmann manifold was proposed in [40]. It was shown that this way of modeling data with union of subspaces instead of a single subspace significantly improves the recognition performance [40].

### B. Feature Transformation-based Approaches

One of the earliest object category adaptation methods was proposed by Saenko *et al.* in [41]. The idea behind this method is to adapt features across general image domains by learning transformations. Given feature vectors  $\mathbf{x}^s \in \mathcal{S}$  and  $\mathbf{x}^t \in \mathcal{T}$ , a linear transformation  $\mathbf{W} \in \mathbb{R}^{N \times M}$  from  $\mathcal{T}$  to  $\mathcal{S}$  is learned.

The inner product similarity function between  $\mathbf{x}^s$  and the transformed  $\mathbf{x}^t$  is denoted by

$$\text{sim}_{\mathbf{W}} = (\mathbf{x}^s)^T \mathbf{W} \mathbf{x}^t. \quad (5)$$

One can view this function as an inner product between the transformed target point  $\mathbf{W} \mathbf{x}^t$  and  $\mathbf{x}^s$ . The objective is to learn the linear transformation given some form of supervision, and then to utilize the learned similarity function in a classification algorithm [41]. A regularization function for the matrix  $\mathbf{W}$  is introduced to avoid over-fitting, which is denoted as  $r(\mathbf{W})$ . Assume that the supervision is a function of the learned similarity values  $\text{sim}_{\mathbf{W}}$ , so a general optimization problem would seek to minimize the regularizer subject to supervision constraints given by functions  $c_i$  :

$$\min_{\mathbf{W}} r(\mathbf{W}) \quad \text{s.t.} \quad c_i(\mathbf{S}^T \mathbf{W} \mathbf{T}) \geq 0, \quad 1 \leq i \leq J. \quad (6)$$

The above problem (6) can be written as an unconstrained problem

$$\min_{\mathbf{W}} r(\mathbf{W}) + \lambda \sum_i c_i(\mathbf{S}^T \mathbf{W} \mathbf{T}). \quad (7)$$

The regularizer studied in [41] is

$$r(\mathbf{W}) = \text{trace}(\mathbf{W}) - \log \det(\mathbf{W}) \quad (8)$$

and the resulting optimization problem is solved using an information-theoretic metric learning (ITML) [42] type of algorithm. One of the limitations of this method is that it can only be applied when the dimensionalities of the two domains are the same (e.g.  $N = M$ ).

This work was extended in [43] by Kulis *et al.* to the more general case where the domains are not restricted to be the same dimensionality and arbitrary asymmetric transformations can be learned. Their method can deal with more general types of domains shift and changes in feature type and dimension. Furthermore, they show that the method in [41] is a special case of their general formulation, producing symmetric positive definite transformations [43]. It was shown that asymmetric indefinite transformations are more flexible for a variety of adaptation tasks than the symmetric transformations.

Recently, a low-rank approximation based approach for semi-supervised domain adaptation was proposed in [44]. The basic goal of this method is to map the source data by a matrix  $\mathbf{W} \in \mathbb{R}^{N \times N}$  to an intermediate representation where each transformed sample can be reconstructed by a linear combination of the target data samples

$$\mathbf{W} \mathbf{S} = \mathbf{T}_l \mathbf{Z}, \quad (9)$$

where  $\mathbf{Z} \in \mathbb{R}^{N_{tu} \times N_s}$  is the coefficient matrix. The following formulation is proposed to solve for the low-rank solution

$$\begin{aligned} (\hat{\mathbf{W}}, \hat{\mathbf{Z}}, \hat{\mathbf{E}}) &= \min_{\mathbf{W}, \mathbf{Z}, \mathbf{E}} \text{rank}(\mathbf{Z}) + \lambda \|\mathbf{E}\|_{2,1}, \\ \text{s.t. } \mathbf{W}\mathbf{S} &= \mathbf{T}_l \mathbf{Z} + \mathbf{E}, \quad \mathbf{W}\mathbf{W}^T = \mathbf{I}, \end{aligned} \quad (10)$$

where  $\text{rank}(\cdot)$  denotes the rank of a matrix,  $\lambda$  is a parameter,  $\mathbf{E} \in \mathbb{R}^{N \times N_s}$  is the error term and the  $\ell_{2,1}$ -norm is defined as  $\|\mathbf{E}\|_{2,1} = \sum_{i=1}^N \sqrt{\sum_{j=1}^{N_s} E_{ij}^2}$ . Since the rank minimization problem (10) is NP-hard, the rank constraint is relaxed by the nuclear norm constraint [44]. The Augmented Lagrange Multiplier (ALM) method is proposed to solve the optimization problem.

Once the solution  $(\hat{\mathbf{W}}, \hat{\mathbf{Z}}, \hat{\mathbf{E}})$  is obtained, the source data is transformed to the target domain as

$$\hat{\mathbf{W}}\mathbf{S} - \hat{\mathbf{E}}. \quad (11)$$

The transformed source data are mixed with the target samples as the augmented training samples for training the classifiers. The trained classifier is then used to perform recognition on the unseen test samples in the target domain [44]. Extension of this method for the multiple source domain adaptation problem has also been proposed in [44].

Other recent transformation-based visual domain adaptation methods include [45] and [46].

### C. Parameter Adaptation Methods

Several algorithms have been proposed in the literature that investigate modifying the Support Vector Machine (SVM) algorithms for the domain adaptation problem. In particular, Yang *et al.* proposed an Adaptive SVM (A-SVM) [29] method in which the source classifier  $f_S(\mathbf{x})$  trained on the source data  $\mathcal{S} = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$ , is adapted to a new classifier  $f_T(\mathbf{x})$  for the unseen target data  $\mathcal{T}_u = \{\mathbf{x}_i^{tu}\}_{i=1}^{N_{tu}}$ . The decision function is formulated as

$$f_T(\mathbf{x}) = f_S(\mathbf{x}) + \delta f(\mathbf{x}), \quad (12)$$

where  $\delta f(\mathbf{x})$  is the perturbation function. It was shown in [29] that the perturbation function can be formulated as  $\delta f(\mathbf{x}) = \boldsymbol{\theta}^T \phi(\mathbf{x})$ , where a feature map  $\phi$  is used to project  $\mathbf{x}$  into a high-dimensional feature vector  $\phi(\mathbf{x})$ . The perturbation function  $\delta f(\mathbf{x})$  is learned using the labeled data  $\mathcal{T}_l = \{(\mathbf{x}_i^{tl}, y_i^{tl})\}_{i=1}^{N_{tl}}$  from the target domain. To learn the parameter  $\mathbf{w}$  of the perturbation function  $\delta f(\mathbf{x})$ , the following optimization

problem is solved

$$\begin{aligned} & \min_{\boldsymbol{\theta}} \frac{1}{2} \|\boldsymbol{\theta}\|^2 + \alpha \sum_{i=1}^{N_{tl}} \xi_i \\ & s.t. \xi_i \geq 0, \\ & y_i^{tl} f_S(\mathbf{x}_i^{tl}) + y_i^{tl} \boldsymbol{\theta}^T \phi(\mathbf{x}_i^{tl}) \geq 1 - \xi_i, \forall (\mathbf{x}_i^{tl}, y_i^{tl}) \in \mathcal{T}_l, \end{aligned} \quad (13)$$

where  $\xi_i$  is the penalizing variable and  $\alpha$  is a parameter that determines how much error an SVM can tolerate. The first term in (13) tries to minimize the deviation between the new decision boundary and the old one, and the second term controls the penalty of the classification error over the training data in the target domain.

This work was improved in [47] for object category detection and in [48] for visual concept classification. Domain transfer SVM [49] attempts to reduce the mismatch in the domain distributions, measured by the maximum mean discrepancy, while also learning a target decision function. Other SVM-based domain adaptation methods include [50], [51], [52], [53], [54].

As discussed in previous sections, several domain adaptation methods make use of the kernel methods. The classification performance of these kernel-based methods is highly dependent on the choice of the kernel. Multiple Kernel Learning (MKL) can be used to combine multiple kernel functions to obtain a better solution [55]. Multiple kernel learning has been shown to work well in many computer vision applications. However, these methods assume that both training and test data come from the same domain. As a result, MKL methods cannot learn the optimal kernel with the combined data from the source and target domains for the domain adaptation problem. Hence, training data from the auxiliary domain may degrade the performance of MKL algorithms in the target domain. To deal with this, several cross domain kernel learning methods have been proposed in the literature [56], [57], [58].

In [56], adaptive multiple kernel learning is utilized to learn a kernel function based on multiple base kernels. In [57], a kernel function and a classifier are simultaneously learned by minimizing both the structural risk functional and the distribution mismatch between the labeled and unlabeled samples from the auxiliary and target domains. It was shown in [56], [57] that these domain adaptive MKL methods can significantly outperform traditional MKL and cross-domain learning methods.

There are some limitations of the feature-based and parameter transfer-based visual domain adaptation methods reviewed in this survey. For instance, the transform-based approaches discussed in [41], [43], [45], [46] are based on some notion of closeness between the transformed source samples and target samples. They do not optimize the objective function of a discriminative classifier directly. Also, the

computational complexity of these methods is highly dependent on the total number of samples used for training. On the other hand, parameter adaptation-based methods such as [29], [48] optimize the classifier directly but they are not able to transfer the adapted function to novel categories. To deal with this problem, several methods have been developed in the literature that attempt to optimize both the transformation and classifier parameters jointly [59], [60], [61].

In particular, Max-Margin Domain Transfer (MMDT) method was recently proposed by Hoffman *et al.* in [60] which uses an asymmetric transform  $\mathbf{W}$  to map target features to a new representation where they are maximally aligned with the source and learns the transform jointly on all categories for which target labels are available. It provides a way to adapt max-margin classifiers in a multi-class setting, by learning a common component of the domain shift as captured by  $\mathbf{W}$ .

The goal of this method is to jointly learn affine hyperplanes that separate the classes in the source domain and a transformation from the points in the target domain into the source domain, such that the transformed target data lie on the correct side of the learned source hyperplanes. For simplicity, let us consider the optimization for the binary problem [60]

$$\begin{aligned} & \min_{\mathbf{W}, \boldsymbol{\theta}, b} \frac{1}{2} \|\mathbf{W}\|_F^2 + \frac{1}{2} \|\boldsymbol{\theta}\|_F^2 \\ & \text{s.t. } y_i^s \left( \begin{bmatrix} \mathbf{x}_i^s \\ 1 \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\theta} \\ b \end{bmatrix} \right) \geq 1 \quad \forall i \in \{1, \dots, N_s\} \\ & \quad y_i^{tl} \left( \begin{bmatrix} \mathbf{x}_i^{tl} \\ 1 \end{bmatrix}^T \mathbf{W}^T \begin{bmatrix} \boldsymbol{\theta} \\ b \end{bmatrix} \right) \geq 1 \quad \forall i \in \{1, \dots, N_{tl}\}, \end{aligned} \quad (14)$$

where  $\boldsymbol{\theta}$  denotes the normal of the affine hyperplane and  $b$  is the bias term. This formulation can be easily extended to the multi-class case by adding a sum over the regularizers on all class-specific parameters and adding the constraints for all categories. The resulting optimization problem is not convex. As a result, it is solved by alternating minimization on  $\mathbf{W}$  and  $(\boldsymbol{\theta}, b)$  [60]. This work was extended in [61] to include Laplacian regularization using instance constraints that are encoded by an arbitrary graph.

Another approach to simultaneous learning of domain-invariant features and classifiers was proposed by Shi and Sha in [59]. Their framework is based on the notion of discriminative clustering in which both the source and target domains are assumed to be tightly clustered and clusters are assumed to correspond to class boundaries. It is assumed that for the same class, the clusters from the two domains are geometrically close to each other. Their formulation of learning the optimal feature space is based on maximizing the domain similarity that makes the source and the target domains look alike and minimizing

the expected classification error on the target domain. An information theoretic framework is proposed for solving their formulation [59].

#### D. Dictionary-based Approaches

The study of sparse representation of signals and images has attracted tremendous interest over the last few years. This is partly due to the fact that signals or images of interest, though high dimensional, can often be coded using few representative atoms in some dictionary. Olshausen and Field in their seminal work [62] introduced the idea of learning dictionary from data instead of using off-the-shelf bases. Since then, data-driven dictionaries have been shown to work well for both image restoration and classification tasks [63], [64]. The efficiency of dictionaries in these wide range of applications can be attributed to the robust discriminant representations that they provide by adapting to particular data samples. However, the learned dictionary may not be optimal if the target data has different distribution than the data used for training. Several dictionary learning-based methods have been proposed in the literature to deal with this domain shift problem [65], [66], [67], [68].

A function learning framework for the task of transforming a dictionary learned from one visual domain to the other, while maintaining a domain-invariant sparse representation of a signal was proposed in [65]. Domain dictionaries are modeled by a linear or non-linear parametric function. The dictionary function parameters and domain-invariant sparse codes are then jointly learned by solving an optimization problem. Motivated by the manifold-based incremental learning work of Gopalan *et al.* [33], [34], Ni *et al.* [67] proposed an unsupervised domain adaptive dictionary learning framework by generating a set of intermediate dictionaries which smoothly connect the source and target domains. One of the important properties of this approach is that it allows the synthesis of data associated with the intermediate domains while exploiting the discriminative power of generative dictionaries. The intermediate data can then be used to build a classifier for recognition under domain shifts.

In [66] Shekhar *et al.* proposed a semi-supervised domain adaptive dictionary learning framework for learning a single dictionary to optimally represent both source and target data. As the features may not be correlated well in the original space, they propose to project data from both the domains onto a common low-dimensional space while maintaining the manifold structure of the data. They argue that learning the dictionary on a low-dimensional space makes the algorithm faster and irrelevant information in the original features can be discarded. Moreover, joint learning of dictionary and projections ensures that the common internal structure of data in both the domains is extracted, which can be represented well by sparse linear combinations of dictionary atoms. Fig. 4 shown an overview of this method [66].



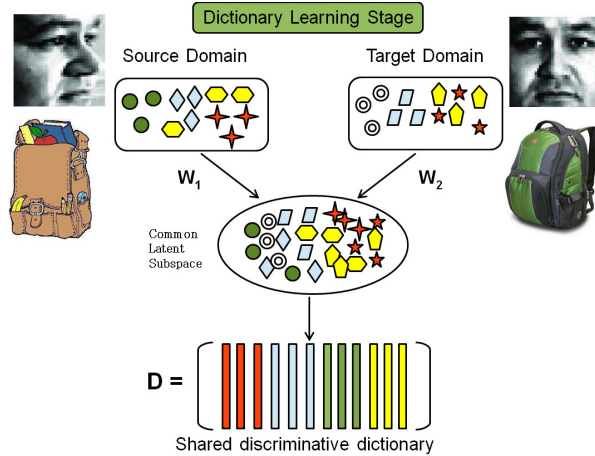


Fig. 4: Overview of the domain adaptive latent space dictionary learning framework [66].

Given source and target domain data  $\mathbf{S} \in \mathbb{R}^{N \times N_s}$  and  $\mathbf{T}_l \in \mathbb{R}^{M \times N_{tl}}$ , respectively, Shekhar *et al.* learn a shared  $K$  atom dictionary,  $\mathbf{D} \in \mathbb{R}^{l \times K}$  and mappings  $\mathbf{W}_1 \in \mathbb{R}^{l \times N}$  and  $\mathbf{W}_2 \in \mathbb{R}^{l \times M}$  onto a common low-dimensional space, which will minimize the representation error in the projected space. Formally, the following cost is minimized

$$\begin{aligned} \mathcal{C}_1(\mathbf{D}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{X}_1, \mathbf{X}_2) = & \|\mathbf{W}_1 \mathbf{S} - \mathbf{D} \mathbf{X}_1\|_F^2 + \\ & \|\mathbf{W}_2 \mathbf{T}_l - \mathbf{D} \mathbf{X}_2\|_F^2 \end{aligned}$$

subject to sparsity constraints on  $\mathbf{X}_1 \in \mathbb{R}^{K \times N_s}$  and  $\mathbf{X}_2 \in \mathbb{R}^{K \times N_{tl}}$ . It is assumed that rows of the projection matrices,  $\mathbf{W}_1$  and  $\mathbf{W}_2$  are orthogonal and normalized to unit-norm. This prevents the solution from becoming degenerate, leads to an efficient scheme for optimization and makes the kernelization of the algorithm possible. Note that this method does not require the data to be of same dimension in source and target domains. As a result, this method is applicable to heterogeneous domain adaptation problems [32].

In order to make sure that the projections do not lose too much information available in the original domains after projecting onto the latent space, a PCA-like regularization term is added which preserves energy in the original signal, given as

$$\begin{aligned} \mathcal{C}_2(\mathbf{W}_1, \mathbf{W}_2) = & \|\mathbf{S} - \mathbf{W}_1^T \mathbf{W}_1 \mathbf{S}\|_F^2 + \\ & \|\mathbf{T}_l - \mathbf{W}_2^T \mathbf{W}_2 \mathbf{T}_l\|_F^2. \end{aligned}$$

It is easy to show that the costs  $\mathcal{C}_1$  and  $\mathcal{C}_2$ , after ignoring the constant terms in  $\mathbf{Y}$ , can be written as

$$\mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}) = \|\tilde{\mathbf{W}}\tilde{\mathbf{Y}} - \mathbf{D}\tilde{\mathbf{X}}\|_F^2, \quad (15)$$

$$\mathcal{C}_2(\tilde{\mathbf{W}}) = -\text{trace}((\tilde{\mathbf{W}}\tilde{\mathbf{Y}})(\tilde{\mathbf{W}}\tilde{\mathbf{Y}})^T) \quad (16)$$

where,

$$\tilde{\mathbf{W}} = [\mathbf{W}_1 \ \mathbf{W}_2], \quad \tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_l \end{pmatrix}, \quad \text{and} \quad \tilde{\mathbf{X}} = [\mathbf{X}_1 \ \mathbf{X}_2].$$

Hence, the overall optimization is given as

$$\begin{aligned} \{\mathbf{D}^*, \tilde{\mathbf{W}}^*, \tilde{\mathbf{X}}^*\} &= \underset{\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}}{\text{argmin}} \mathcal{C}_1(\mathbf{D}, \tilde{\mathbf{W}}, \tilde{\mathbf{X}}) + \lambda \mathcal{C}_2(\tilde{\mathbf{W}}) \\ \text{s.t. } &\mathbf{W}_i \mathbf{W}_i^T = \mathbf{I}, \quad i = 1, 2 \quad \text{and} \quad \|\tilde{\mathbf{x}}_j\|_0 \leq T_0, \forall j \end{aligned} \quad (17)$$

where,  $\lambda$  is a positive constant. An efficient two-step procedure is proposed for solving this optimization problem in [66]. Furthermore, this method has been extended to multiple domains and kernelized in [66]. Once the projection matrices and the dictionary are learned, given a novel test sample from the target domain, it is first projected onto the latent domain using  $\mathbf{W}_2$  and classified using a variation of the Latent Sparse Embedding Residual Classifier (LASERC) algorithm proposed in [69].

### E. Domain Resampling

An unsupervised domain adaptation method was recently proposed in [70], [71] based on the notion of landmarks. Landmarks are a subset of labeled data instances in the source domain that are distributed most similarly to the target domain [70]. The key insight of their method is that not all instances are created equally for adaptation. As a result, they pick out and exploit the most desirable instances to facilitate adaptation. An overview of this method is shown in Fig. 5.

A variant of Maximum Mean Discrepancy (MMD) is used to select samples from the source domain to match the distribution of the target domain. To identify landmarks,  $N_s$  indicator variables  $\alpha = \{\alpha_i \in \{0, 1\}\}$  are used, one for each data point in the source domain. If  $\alpha_i = 1$ , then  $\mathbf{x}_i^s$  is regarded as a landmark. The vector  $\alpha$  is identified by minimizing the MMD metric, defined with a kernel mapping function  $\phi(\mathbf{x})$ ,

$$\begin{aligned} \min_{\alpha} &\left\| \frac{1}{\sum_i \alpha_i} \sum_i \alpha_i \phi(\mathbf{x}_i^s) - \frac{1}{N_{tu}} \sum_j \phi(\mathbf{x}_j^{tu}) \right\|_{\mathcal{H}}^2 \\ \text{s.t. } &\frac{1}{\sum_i \alpha_i} \sum_i \alpha_i y_{ic} = \frac{1}{N_s} \sum_i y_{ic}, \end{aligned} \quad (18)$$

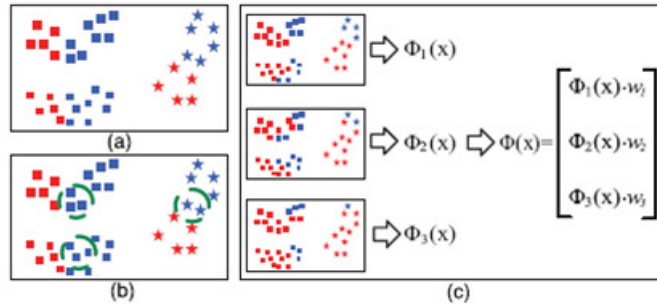


Fig. 5: An overview of the landmark-based method proposed in [70]. (a) The original domain adaptation problem where instances in red are from the target and in blue from the source. (b) Landmarks, shown inside the green circles, are data instances from the source that can be regarded as samples from the target. (c) Multiple auxiliary tasks are created by augmenting the original target with landmarks, which switches their color from blue to red. Each task gives rise to a new feature representation. These representations are combined discriminatively to form domain-invariant features for the original domain adaptation problem [70].

where  $y_{ic}$  is the indicator variable for  $y_{ic} = c$ . The right hand side of the constraint is simply the prior probability of the class  $c$ , estimated from the source domain.

The geodesic flow kernel computed between the source  $\mathcal{S}$  and the target  $\mathcal{T}_u$  is used to compose the kernel mapping function  $\phi(\mathbf{x})$  [70]

$$\begin{aligned} \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) &= K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \exp\{-(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{G} (\mathbf{x}_i - \mathbf{x}_j) / \sigma^2\}, \end{aligned} \quad (19)$$

where  $\mathbf{G}$  is computed using the singular value decomposition of  $\mathbf{S}_1^T \mathbf{S}_2$ . Here,  $\mathbf{S}_1$  and  $\mathbf{S}_2$  are the matrices obtained by applying PCA on  $\mathbf{S}$  and  $\mathbf{T}_u$ , respectively [37].

A set of factors  $\{\sigma_i \in [\sigma_{\min}, \sigma_{\max}]\}_{i=1}^Q$  is used to select the scale factor  $\sigma$  in (19). For each  $\sigma_i$ , (18) is solved to obtain the corresponding landmarks  $\mathcal{L}^i$  whose  $\alpha_i$  is equal to 1. For each set of landmarks, a new domain pair is constructed by moving the landmarks from the original source to the target domains. It was argued that each auxiliary task is easier to adapt than the original pair  $\mathcal{S}$  and  $\mathcal{T}_u$  [70].

The final kernel is then learned as a convex combination of all the kernels from the auxiliary tasks

$$\mathbf{F} = \sum_i \beta_i \mathbf{G}_i \quad \text{s.t.} \quad \beta_i \geq 0 \quad \text{and} \quad \sum_i \beta_i = 1. \quad (20)$$

The coefficients  $\beta_i$  are optimized on a labeled training set  $\sum_i \mathcal{L}^i$  composed of all landmarks selected at different granularities. Finally,  $\mathbf{F}$  is used in a SVM classifier whose accuracy is optimized with the standard MKL algorithm to learn  $\beta_i$  [70], [71]. Since  $\sum_i \mathcal{L}^i$  consists of landmarks that are distributed similarly to the target, it is expected that the classification error on  $\sum_i \mathcal{L}^i$  to be a good proxy to that of the target domain [70].

### F. Other Methods

Deep neural networks have had tremendous success achieving state-of-the-art performance on a number of machine learning and computer vision tasks [72]. This is due in part to the fact that deep networks are able to learn extremely powerful hierarchical non-linear representations of the inputs [73], [74]. Motivated by recent works on deep learning, several hierarchical domain adaptation approaches have been proposed in the literature [75], [76], [77], [78], [79].

In [78], multiple intermediate representations are explored along an interpolating path between the target and source domains. Starting with all the source data samples  $\mathcal{S}$ , intermediate sampled datasets are generated. For each successive dataset, the proportion of samples randomly drawn from  $\mathcal{T}$  is increased and the proportion of samples drawn from  $\mathcal{S}$  is decreased. Let  $i \in [1, \dots, k]$  be an index set over  $k$  intermediate datasets. Then,  $\mathcal{S}_i = \mathcal{S}$  for  $i = 1$ ,  $\mathcal{S}_i = \mathcal{T}$  for  $i = k$ . For  $i \in [2, \dots, k - 1]$ , datasets  $\mathcal{S}_i$  and  $\mathcal{S}_{i+1}$  are created in a way so that the proportion of samples from  $\mathcal{T}$  in  $\mathcal{S}_i$  is less than in  $\mathcal{S}_{i+1}$ . Each of these datasets can be thought of as a single point on a particular kind of interpolating path between  $\mathcal{S}$  and  $\mathcal{T}$ .

For each intermediate dataset  $\mathcal{S}_i$ , a deep non-linear feature extractor is trained. Once feature extractors corresponding to all points on the path are trained, any input sample can be represented by concatenating all the outputs from the feature extractors together to create path features for the input. The hope is that this path representation will be more effective at domain adaptation because it is constructed to capture information about incremental changes between the source and target domains similar to [33], [37]. After creating the path representation of the inputs, a classifier is trained on the data generated from the source domain data by minimizing an appropriate loss function [78].

Another recent work for visual domain adaptation using hierarchical networks was recently proposed by Nguyen *et al.* in [77]. Their method jointly learns a hierarchy of features together with transformations that address the mismatch between different domains. This method was motivated by [80] in which multi-layer sparse coding networks are proposed for building feature hierarchies layer by layer using sparse codes and spatial pooling. Fig. 6 shows an overview of the sparse hierarchical domain adaptation method

[77]. The network contains multiple layers, each of which contains 3 sub-layers. The first sub-layer performs contrast-normalization and dimensionality reduction on the input data. Sparse coding is carried out in the second sub-layer. In the final sub-layer, adjacent features are max-pooled together to produce a new features. Output from one layer becomes the input to the next layer. This method can be viewed as a generalization of the domain adaptive dictionary learning framework [66] using hierarchical networks. Extension of this method to multiple source domains has also been presented in [77].

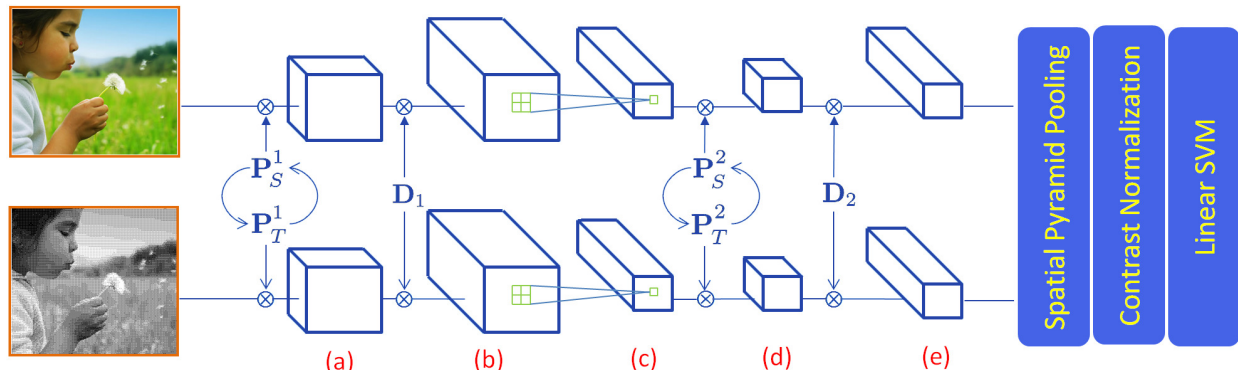


Fig. 6: An illustration of Domain Adaptation using a Sparse and Hierarchical Network (DASH-N) algorithm [77]. The source domain is RGB images and the target domain is halftone images. First, images are divided into small overlapping patches. These patches are vectorized while maintaining their spatial arrangements. (a) Performing contrast-normalization and dimensionality reduction using  $\mathbf{P}_S$  for source images and  $\mathbf{P}_T$  for target images. The circular feedbacks between  $\mathbf{P}_S$  and  $\mathbf{P}_T$  indicate that these two transformations are learned jointly. (b) Obtaining sparse codes using the common dictionary  $\mathbf{D}_1$ . (c) Performing max pooling. The process then repeats for layer 2 (d & e), except that the input is the sparse codes from layer 1 instead of pixel intensities. At the final stage, spatial pyramid with max pooling are used to create image descriptors. Classification is done using linear support vector machine.

Visual attributes are human understandable properties to describe images such as blue, dark, two-legged. They are valuable as a semantic cue in various vision problems. Recent research explores a variety of applications for visual attributes including face verification [81], object recognition [82], [83], [84] and facilitating transfer learning [85]. Existing methods [85], [82], [84] assume that one model of an attribute is sufficient to capture all user perceptions. However, there are some real perceptual differences between annotators. Consider the example shown in Fig. 7, five users confidently declared the shoe on the left is formal, while five confidently declared the opposite. These differences stem from several factors such

as the words for attributes are imprecise, their meaning often depends on context and culture and they often stretch to refer to quite distinct object categories [86].

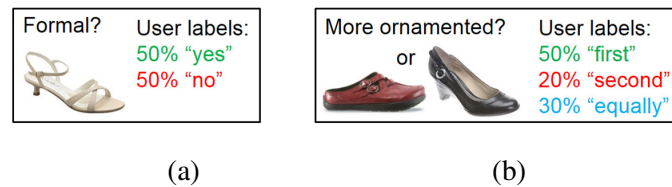


Fig. 7: Virtual attribute interpretations vary slightly from viewer to viewer. For instance, five viewers confidently declare the shoe as formal (a) or more ornamented (b), while five others confidently declare the opposite. Attribute adaptation models are proposed to take these differences in perception into account [86].

In order to capture the inherent differences in perception, [86] proposes to model attributes in a user-specific way. In particular, attribute learning is posed as an adaptation problem. First, they leverage any commonalities in perception to learn a generic prediction function using a large margin learning algorithm and data labeled with majority vote from multiple annotators. Then, they use a small number of user-labeled examples to adapt the parameters of the generic model into a user-specific prediction function, while not straying too far from the prior generic model. Essentially this amounts to imposing regularizers on the learning objective favoring user-specific model parameters that are similar to the generic ones, while still satisfying the user-specific label constraints [86]. The impact of this attribute adaptation work is that one can capture a user’s perception with minimal annotation effort. It was shown that the resulting personalization can make attribute-based image search more accurate [86].

Tommasi and Caputo [87] very recently proposed a Naive Bayes Nearest Neighbor (NBNN)-based domain adaptation method that iteratively learns a Mahalanobis class specific metric, while inducing for each sample a large margin separation among classes. Both semi-supervised and unsupervised domain adaptation scenarios are presented.

In [88] Jain and Farfade proposed an approach for adapting a cascade of classifiers to perform classification in a similar domain for which only a few positive examples are available. A cascade of classifiers is a classifier  $f$  that is composed of  $m$  stage classifiers  $\{f_1, \dots, f_m\}$  that are applied in a sequential manner. They are commonly used for anomaly detection and one-class classification. It was shown that by adapting adapting classification cascades to new domains one can obtain huge gains in performance in detecting faces of human babies and human-like characters from movies.

## IV. APPLICATIONS

In this section, we illustrate through different application examples the use and capabilities of various visual domain adaptation methods. In particular, we focus on object recognition and face recognition applications.

### A. Face Recognition

Face recognition is a challenging problem that has been actively researched for over two decades [89]. Current systems work very well when training and test images are captured under controlled conditions. However, their performance degrades significantly when the test images contain variations that are not present in the training images. One of these variations is change in pose. Along with the frontal images with different illumination (source images), if we are also given a few images at different poses (target images), then the resulting face recognition problem can be viewed a domain adaptation problem [66], [65], [90].

Face recognition experiments were conducted on the CMU Multi-PIE dataset [91] with images of 129 subjects in frontal pose as the source domain, and five other off-frontal poses as the target domain. Images under five illumination conditions across source and target domains were used for training with which images from remaining 15 illumination conditions in the target domain were recognized. Results provided in Table I show that the dictionary-based adaptation method [66] compares favorably with some of the recently proposed multi-view recognition algorithms [10] as well as many other non-adaptation techniques, and gives the best performance on average. Note that the discriminative dictionary learning algorithm, FDDL [92] does not provide the best results here as it is not able to efficiently represent the non-linear changes introduced by the pose variation.

Furthermore, the learned dictionaries were also used for pose alignment where the goal is to align faces from one pose to a different pose. This is a challenging problem since actual pose variations are three dimensional whereas the image evidence one has is two dimensional. Sample results are shown in Figure 8. One of the interesting features of the dictionary-based adaptation methods is that they allow the synthesis of data associated with different domains while exploiting the generative power of dictionary-based representations. This is essentially what is highlighted in the last two rows of Figure 8. The dictionary-based method is robust at high levels of noise and missing pixels. It produces denoised and inpainted synthesized images. Additional results on various face recognition tasks using domain adaptation can be found in [65] and [67].

Method	Probe pose					Average
	15°	30°	45°	60°	75°	
PCA	15.3	5.3	6.5	3.6	2.6	6.7
PLS [27]	39.3	40.5	41.6	41.1	38.7	40.2
LDA	98.0	94.2	91.7	84.9	79.0	89.5
CCA [27]	92.1	89.7	88.0	86.1	83.0	83.5
GMLDA [10]	<b>99.7</b>	<b>99.2</b>	98.6	94.9	95.4	97.6
FDDL [92]	96.8	90.6	94.4	91.4	90.5	92.7
SDDL [66]	98.4	98.2	<b>98.9</b>	<b>99.1</b>	<b>98.8</b>	<b>98.7</b>

TABLE I: Comparison of various algorithms for face recognition across pose [66].

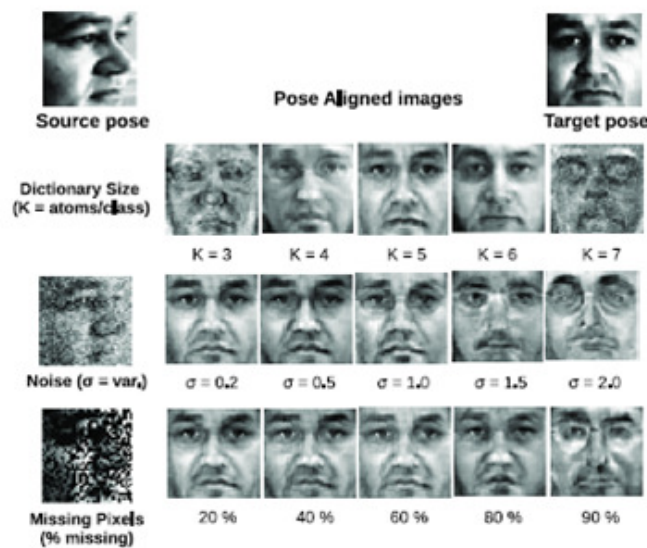


Fig. 8: Examples of pose-aligned images. Synthesis in various conditions demonstrate the robustness of the domain adaptive dictionary learning method [66].

### B. Object Recognition

In this section, we compare the performance of various visual domain adaptation methods on a benchmark object recognition dataset which was introduced in [41]. The dataset consists of images from three sources: Amazon (consumer images from online merchant sites), DSLR (images by DSLR camera) and Webcam (low quality images from webcams). In addition, algorithms are tested on the Caltech-256



dataset [93], taking it as the fourth domain. Fig. 9 shows sample images from these datasets, and clearly highlights the differences between them.



Fig. 9: Example images from KEYBOARD and BACK-PACK categories in Caltech-256, Amazon, Webcam and DSLR. Caltech-256 and Amazon datasets have diverse images, Webcam and DSLR are similar datasets with mostly images from offices [66].

Three set-ups are followed for comparing the performance of various algorithms. In the first set-up, 10 classes: BACKPACK, TOURING-BIKE, CALCULATOR, HEADPHONES, COMPUTER- KEYBOARD, LAPTOP-101, COMPUTER- MONITOR, COMPUTER-MOUSE, COFFEE- MUG, AND VIDEO- PROJECTOR, common to all the four datasets are used. In this case, there are a total of 2533 images. Each category has 8 to 151 images in a dataset. In the second set-up, all 31 classes from Amazon, Webcam and DSLR are used to evaluate various algorithms. Finally, in the third set-up, methods for adaptation are evaluated using multiple domains. In this case, the first dataset is used and methods are tested on all the 31 classes in it. For both the cases, we use 20 training samples per class for Amazon/Caltech, and 8 samples per class for DSLR/Webcam when used as source, and 3 training samples for all of them when used for target domain. Rest of the data in the target domain is used for testing. The experiment is run multiple times for random train/test splits and the result is averaged over all the runs. For the unsupervised case, the same setting as semi-supervised adaptation described above is followed but without using any labeled

data from the target domain<sup>2</sup>.

1) *Semi-supervised Adaptation Results using Single Source*: The semi-supervised adaptation recognition results of different algorithms on 8 pairs of source-target domains and on all 31 classes are shown in Table II and Table III, respectively. Baseline results obtained using the Hierarchical Matching Pursuit (HMP) method [80] as well as the Fisher Discrimination Dictionary Learning (FDDL) method [92] which learn the dictionaries separately for the source and target domains without performing domain adaptation are also included.

Methods	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
Metric [41]	33.7 ± 0.8	35.0 ± 1.1	27.3 ± 0.7	36.0 ± 1.0	21.7 ± 0.5	32.3 ± 0.8	30.3 ± 0.8	55.6 ± 0.7
SGF [33]	40.2 ± 0.7	36.6 ± 0.8	37.7 ± 0.5	37.9 ± 0.7	29.2 ± 0.7	38.2 ± 0.6	39.2 ± 0.7	69.5 ± 0.9
GFK [37]	46.1 ± 0.6	55.0 ± 0.9	39.6 ± 0.4	56.9 ± 1.0	32.8 ± 0.1	46.2 ± 0.6	46.2 ± 0.6	<b>80.2 ± 0.4</b>
FDDL [92]	39.3 ± 2.9	55.0 ± 2.8	24.3 ± 2.2	50.4 ± 3.5	22.9 ± 2.6	41.1 ± 2.6	36.7 ± 2.5	65.9 ± 4.9
HMP [80]	67.7 ± 2.3	70.2 ± 5.1	51.7 ± 4.3	70.0 ± 4.2	46.8 ± 2.1	61.5 ± 3.8	64.7 ± 2.0	76.0 ± 4.0
SDDL [66]	49.5 ± 2.6	76.7 ± 3.9	27.4 ± 2.4	72.0 ± 4.8	29.7 ± 1.9	49.4 ± 2.1	48.9 ± 3.8	72.6 ± 2.1
DASH-N [77]	<b>71.6 ± 2.2</b>	<b>81.4 ± 3.5</b>	<b>54.9 ± 1.8</b>	<b>75.5 ± 4.2</b>	<b>50.2 ± 3.3</b>	<b>70.4 ± 3.2</b>	<b>68.9 ± 2.9</b>	77.1 ± 2.8

TABLE II: Semi-supervised domain adaptation results of different approaches on four domains with 10 common classes (C: Caltech, A: Amazon, D: DSLR, W: Webcam).

Compared to the metric learning-based approach [41], manifold-based feature concatenation methods [33], [37] provide better results. This makes sense because by finding intermediate domain representations one is able to learn a feature vector that is more robust than a feature vector that results by learning a single transformation that minimizes the effect of the domain shift. The SDDL method can be viewed as an extension of the FDDL method which simultaneously learns discriminative dictionaries on a latent space where both the source and the target data are forced to have similar sparse representation. As a result, one can clearly see the performance gain of the SDDL method over the FDDL method as well as the manifold-based methods in Table II and Table III.

The HMP method [80] builds a feature hierarchy layer by layer using an efficient matching pursuit encoder. It consists of three main components: batch tree orthogonal matching pursuit, spatial pyramid matching, and contrast normalization. As a results, it is robust to some of the variations present in the images such as illumination changes, pose variations and resolution variations. The DASH-N method

<sup>2</sup>Several recent methods explore both source and target data at once in a transductive manner rather than splitting the datasets into multiple training/testing partitions. See [70] for details on the evaluation protocol using this setting.

essentially extends the SDDL and HMP methods by learning features directly from data for domain adaptation. As a result, it provides more robust and discriminative representation of the data and performs the best on this dataset on both settings. The dictionary learning-based methods [92], [80] essentially find the common internal structure of the data. They inherently have the denoising capability and provide robust representation of the data. This is one of the reasons why in some cases the FDDL and the HMP methods provide better results than metric learning and manifold-based methods.

Method	A $\rightarrow$ W	D $\rightarrow$ W	W $\rightarrow$ D
Metric [41]	44	31	27
RDALR [44]	50.7 $\pm$ 0.8	36.9 $\pm$ 19.9	32.9 $\pm$ 1.2
SGF [33]	57 $\pm$ 3.5	36 $\pm$ 1.1	37 $\pm$ 2.3
GFK [37]	46.4 $\pm$ 0.5	61.3 $\pm$ 0.4	66.3 $\pm$ 0.4
HMP [80]	55.7 $\pm$ 2.5	50.5 $\pm$ 2.7	56.8 $\pm$ 2.6
SDDL [66]	50.1 $\pm$ 2.5	51.2 $\pm$ 2.1	50.6 $\pm$ 2.6
DASH-N [77]	<b>60.6 <math>\pm</math> 3.5</b>	<b>67.9 <math>\pm</math> 1.1</b>	<b>71.1 <math>\pm</math> 1.7</b>

TABLE III: Single-source semi-supervised domain adaptation results on all 31 classes.

2) *Semi-supervised Adaptation Results using Multiple Sources:* As some of the methods reviewed in this paper can also handle multiple domains, we report results of different algorithms on multiple-source adaption. Table IV shows the results for three possible combinations. Again, the sparse hierarchical network-based adaptation method [77] performs the best. Incremental learning motivated manifold method [34] also provides good results on multi-domain adaptation using this dataset. It is interesting to see that increasing the number of domains can be helpful, especially when compared to a single source and single target. Many multi-domain adaptation methods in Table IV outperform a single source and a single target in many cases, though in a small number of cases they do not outperform a single source and a single target. As a result, a better strategy to deal with multiple domains is required in these cases.

Source	Target	SGF [34]	SGF [33]	RDALR [44]	FDDL [92]	SDDL [66]	A-SVM [29]	HMP [80]	DASH-N [77]
dslr, amazon	webcam	<b>64.5 <math>\pm</math> 0.3</b>	52 $\pm$ 2.5	36.9 $\pm$ 1.1	41.0 $\pm$ 2.4	57.8 $\pm$ 2.4	30.4 $\pm$ 0.6	47.2 $\pm$ 1.9	<b>64.5 <math>\pm</math> 2.3</b>
amazon, webcam	dslr	51.3 $\pm$ 0.7	39 $\pm$ 1.1	31.2 $\pm$ 1.3	38.4 $\pm$ 3.4	56.7 $\pm$ 2.3	25.3 $\pm$ 1.1	51.3 $\pm$ 1.4	<b>68.6 <math>\pm</math> 3.7</b>
webcam, dslr	amazon	38.4 $\pm$ 1.0	28 $\pm$ 0.8	20.9 $\pm$ 0.9	19.0 $\pm$ 1.2	24.1 $\pm$ 1.6	17.3 $\pm$ 0.9	37.3 $\pm$ 1.4	<b>41.8 <math>\pm</math> 1.1</b>

TABLE IV: Multiple-source domain adaptation results of various methods on the Amazon, Webcam and DSLR datasets.

3) *Unsupervised Domain Adaptation Results*: Results on three source-target combinations of the Amazon, DSLR, Webcam datasets are shown in Table V. The manifold-based approach [34] outperforms the existing unsupervised domain adaptation methods in two of the three source-target combinations. Information theoretic learning method [59] for unsupervised domain adaptation also performs well on this dataset. By comparing results in Tables II and III, with results in Table V, we see that the semi-supervised adaptation results are in general better than the unsupervised case. Using labels in both intermediate data generation and classification stage generally produces better results than using labels only during classification [34]. Also, it is interesting to see that since the introduction of this dataset in [41], the recognition performance has significantly improved in the last few years.

Source	Target	SGF [34]	SGF [33]	RDALR [44]	GFK [37]	ITLUDA [59]
webcam	dslr	<b>71.2</b>	$19 \pm 1.2$	$32.89 \pm 1.2$	$49.7 \pm 0.5$	-
dslr	webcam	68.8	$26 \pm 0.8$	$36.85 \pm 1.9$	$44.6 \pm 0.3$	<b><math>83.6 \pm 0.5</math></b>
Amazon	webcam	<b>55.6</b>	$39 \pm 2.0$	$50.71 \pm 0.8$	$15 \pm 0.4$	$38.5 \pm 1.3$

TABLE V: Unsupervised domain adaptation results of various methods on the Amazon, Webcam and DSLR datasets.

### C. Computational Complexity

The main processing steps involved in manifold-based adaptation techniques [33], [37], [34] are computing the geodesic between the source and target domains, and then sampling points along the geodesic to infer intermediate domains that account for the domain shift. This involves mapping entities on the manifold to the locally Euclidean tangent plane, and warping the results from the tangent plane back onto the manifold. Computationally efficient algorithms for these steps have been discussed in the literature for Grassmann manifolds [94]. For orthogonal matrices of dimensions  $N_1 \times N_2$ , the geodesic computation has a complexity of  $O(N_1^2 N_2)$  along with an  $O(N_1 N_2)$  cost for sampling each point along the geodesic.

For deep learning approaches [77], [78], the complexity depends, among others, on the number of layers used in the hierarchy to learn feature correlation for adaptation. While the deep network circuits can have different architectures such as auto-encoders and restricted Boltzmann machines, there is an active stream of work in making the training procedure of these circuits computationally tractable. See [72] for more detailed discussion on the complexity of deep architectures.

Major computational heavy step of dictionary-based domain adaptation methods is dominated by sparse coding. Efficient batch methods have been proposed to learn dictionaries for large scale problems. For instance, a batch orthogonal matching pursuit-based KSVD algorithm for learning dictionaries was proposed in [95]. It was shown that the operation count per training iteration for learning a dictionary of size  $l \times K$  with  $R$  number of training signals are  $R(T_0^2 K + 2lK)$ , where  $T_0$  is the target sparsity. One can also adapt fast  $\ell_1$  solvers for sparse coding [96], [97] rather than using greedy orthogonal matching pursuit algorithms.

For the low-rank approximation-based methods, the major computation is in finding the SVD of a matrix. As a result, these methods tend to be time consuming if the matrix is large. However, efficient methods do exist for finding low-rank approximation of large matrices [98], [99], [100].

Many parameter adaptation methods such as A-SVM [29] are large scale quadratic programming problems for which efficient implementations do exist in the literature. See [101] for more details.

## V. CONCLUSIONS AND FUTURE DIRECTIONS

This article attempted to provide an overview of recent developments in domain adaptation for computer vision, with an emphasis on applications to the problems of face and object recognition. We believe the availability of massive data has brought substantial opportunities and challenges to the analysis of datasets bias or covariant shifts, and domain adaptation problems. We hope that the survey has helped to guide an interested reader among the extensive literature to some degree, but obviously it cannot cover all the literature on domain adaptation, and we have chosen a representative subset of the latest progress made in computer vision to focus on.

Domain adaptation promises to be an active area of research, especially as one of the possible ways to quickly propagate semantic annotations to the large-scale visual data being acquired at every minute. In computer vision, researchers have identified specific challenges that do not belong to machine learning: a major question among them that is rarely addressed in traditional domain adaptation research is one of adapting structured (non-vector) data representations. In machine learning or natural language processing, an input sample is usually represented as a vector in Euclidean space, different samples are treated as independent observations, and the task is typically classification. This is, however, not the case in computer vision where the representations to be potentially adapted include shapes and contours, deformable and articulated 2-D or 3-D objects, graphs and random fields, intrinsic images, as well as visual dynamics, none of which is directly supported by “vectorial” domain adaptation techniques. In addition to recognition and detection, models and algorithms for segmentation, reconstruction, and

tracking are awaiting mechanisms that do not yet exist, to be adapted toward emerging new domains. All of these challenges necessitate continuous efforts on characterizing visual domain shift and a paradigm of effective and efficient adaptation methods that are dedicated to visual data.

In the meantime, it is generally accepted that domain shifts in computer vision are usually due to causes from the imaging process that can be explained physically, such as illumination changes, sensor changes, view point changes, etc.. We believe incorporating these physical priors into strong statistical adaptation approaches will not only lead to performance increase, but also lead to other insights in understanding the imaging process. This calls for a “physically-informed” adaptation paradigm that better exploits knowledge about image formation and better integrates other domain-specific knowledge implied by the diverse set of partial, noisy, and multi-modal “side information” accompanying the visual data, such as imagery obtained from online social media. We hope that by appropriately incorporating physically-informed adaptation paradigm, distributional changes across different sensors (EO/SAR, IR/SAR, Eo/IR, etc.) can be handled.

Finally, we expect that studies on data characteristics and adaptations will produce stronger guidance to developing more desirable datasets for evaluating research in a wider spectrum of computer vision problems.

#### ACKNOWLEDGMENT

This work was partially supported by a MURI from the Office of Naval Research under the grant 1141221258513 and a grant from Xerox Corporation.

#### AUTHORS

**Vishal M. Patel** (pvishalm@umd.edu) is a member of the research faculty at the University of Maryland Institute for Advanced Computer Studies (UMIACS). His research interests are in signal processing, computer vision and machine learning with applications to imaging and biometrics. Dr. Patel was a recipient of the ORAU postdoctoral fellowship in 2010. He is a member of the IEEE, Eta Kappa Nu, Pi Mu Epsilon, and Phi Beta Kappa.

**Raghuraman Gopalan** (raghuram@research.att.com) is a senior member of technical staff at the AT&T Labs-Research. He received his Ph.D. in Electrical and Computer Engineering at the University of Maryland, College Park in 2011. His research interests are in computer vision and machine learning, with a focus on object recognition problems.

**Ruonan Li** (ruonanli@seas.harvard.edu) received the B.E. and M.E. degrees from Tsinghua University, Beijing, China. He received the Ph.D. degree in Electrical and Computer Engineering from the University of Maryland, College Park in 2011. He is currently a research associate at Harvard University. His research interests include general problems in computer vision, image processing, pattern recognition, and machine learning, with recent focuses on video analysis and video based recognition, socialized visual analytics, cross-domain model adaptation, and the application of differential geometric methods to the related problems.

**Rama Chellappa** (rama@umiacs.umd.edu) is a Minta Martin Professor of Engineering and the Chair of the ECE department at the University of Maryland. He is a recipient of the K.S. Fu Prize from IAPR, the Society, Technical Achievement and Meritorious Service Awards from the IEEE Signal Processing Society (SPS) and the Technical Achievement and Meritorious Service Awards from the IEEE Computer Society. At UMD, he has received college and university level recognitions for research, teaching, innovation and mentoring of undergraduate students. He is a Fellow of IEEE, IAPR, OSA, AAAS and ACM and holds four patents.

## REFERENCES

- [1] H. Shimodaira, “Improving predictive inference under covariate shift by weighting the log-likelihood function,” *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.
- [2] A. Torralba and A. A. Efros, “Unbiased look at dataset bias,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
- [3] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, “Undoing the damage of dataset bias,” in *European Conference on Computer Vision*, 2012.
- [4] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” *Intelligent Data Analysis*, vol. 6, no. 5, pp. 429–450, 2002.
- [5] J. J. Heckman, “Sample selection bias as a specification error,” *Econometric*, vol. 47, no. 1, pp. 153–161, 1979.
- [6] B. Zadrozny, “Learning and evaluating classifiers under sample selection bias,” in *International Conference on Machine Learning*, 2004, pp. 114–121.
- [7] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [8] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, “Self-taught learning: transfer learning from unlabeled data,” in *International conference on Machine learning*, 2007, pp. 759–766.
- [9] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. MIT Press, 2006.
- [10] A. Sharma, A. Kumar, H. Daume, and D. Jacobs, “Generalized multiview analysis: A discriminative latent space,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2160–2167.
- [11] J. Jiang, “Domain adaptation in natural language processing,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2008.
- [12] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

- [13] M. Sugiyama and K. R. Muller, "Input-dependent estimation of generalization error under covariate shift," *Statistics & Decisions*, vol. 23, no. 4, pp. 249–279, 2005.
- [14] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. Cambridge, MA: MIT Press, 2007, pp. 601–608.
- [15] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, vol. 46, no. 1-3, pp. 191–202, 2002.
- [16] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [17] W. Dai, G. rong Xue, Q. Yang, and Y. Yu, "Transferring naive Bayes classifiers for text classification," in *AAAI Conference on Artificial Intelligence*, 2007, pp. 540–545.
- [18] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, no. 2-3, pp. 103–134, May 2000.
- [19] D. Xing, W. Dai, G.-R. Xue, and Y. Yu, "Bridged refinement for transfer learning," in *European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2007, pp. 324–335.
- [20] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in nlp," in *Annual Meeting of the Association for Computational Linguistics*, 2007, pp. 264–271.
- [21] R. Raina, "Self-taught learning," Ph.D. dissertation, Stanford University, 2009.
- [22] H. Wang, F. Nie, and H. Huang, "Robust and discriminative self-taught learning," in *International Conference on Machine Learning*, vol. 28, no. 3, 2013, pp. 298–306.
- [23] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Self-taught clustering," in *International Conference on Machine Learning*. ACM, 2008, pp. 200–207.
- [24] Y. Jia, M. Salzman, and T. Darrell, "Factorized latent spaces with structured sparsity," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 982–990.
- [25] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *International Conference on Multimedia*. ACM, 2010, pp. 251–260.
- [26] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *International Conference on Computer Vision*, 2013.
- [27] A. Sharma and D. Jacobs, "Bypassing synthesis: Pls for face recognition with pose, low-resolution and sketch," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 593–600.
- [28] Y. Jia, M. Salzman, and T. Darrell, "Learning cross-modality similarity for multinomial data," in *IEEE International Conference on Computer Vision*, 2011, pp. 2407–2414.
- [29] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive SVMs." in *ACM Multimedia*. ACM, 2007, pp. 188–197.
- [30] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *International Conference on Machine Learning*, 2009, pp. 289–296.
- [31] H. Daumé III, "Frustratingly easy domain adaptation," in *Conference of the Association for Computational Linguistics*, 2007.
- [32] W. Li, L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," pp. 1–1, 2013.



- [33] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *IEEE International Conference on Computer Vision*, 2011, pp. 999–1006.
- [34] —, "Unsupervised adaptation across domain shifts by generating intermediate data representations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, accepted 2013.
- [35] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko, "Discovering latent domains for multisource domain adaptation," in *European Conference on Computer Vision*, 2012, pp. 702–715.
- [36] B. Gong, K. Grauman, and F. Sha, "Reshaping visual datasets for domain adaptation," in *Neural Information Processing Systems*, 2013, pp. 1286–1294.
- [37] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.
- [38] B. Gong, K. Grauman, and F. Sha, "Learning kernels for unsupervised domain adaptation with applications to visual object recognition," *International Journal of Computer Vision*, 2014.
- [39] J. Zheng, M.-Y. Liu, R. Chellappa, and P. Phillips, "A Grassmann manifold-based domain adaptation approach," in *International Conference on Pattern Recognition*, 2012, pp. 2095–2099.
- [40] A. Shrivastava, S. Shekhar, and V. M. Patel, "Unsupervised domain adaptation using parallel transport on Grassmann manifold," in *IEEE Winter conference on Applications of Computer Vision*, 2014.
- [41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell., "Adapting visual category models to new domains," in *European Conference on Computer Vision*, vol. 6314, 2010, pp. 213–226.
- [42] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *International Conference on Machine Learning*, 2007, pp. 209–216.
- [43] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1785–1792.
- [44] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2168–2175.
- [45] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, "Unsupervised domain adaptation by domain invariant projection," in *IEEE International Conference on Computer Vision*, 2013.
- [46] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *IEEE International Conference on Computer Vision*, 2013.
- [47] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *IEEE International Conference on Computer Vision*, 2011, pp. 2252–2259.
- [48] W. Jiang, E. Zavesky, S.-F. Chang, and A. Loui, "Cross-domain learning methods for high-level visual concept classification," in *IEEE International Conference on Image Processing*, 2008, pp. 161–164.
- [49] L. Duan, I. W.-H. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1375–1381.
- [50] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 181–189.
- [51] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

- [52] L. Duan, D. Xu, and I. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 504–518, 2012.
- [53] A. J. Ma, P. C. Yuen, and J. Li, "Domain transfer support vector ranking for person re-identification without target camera label information," in *IEEE International Conference on Computer Vision*, 2013.
- [54] L. Duan, D. Xu, and S.-F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1338–1345.
- [55] M. Gönen and E. Alpaydm, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, Jul. 2011.
- [56] L. Duan, D. Xu, I.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1667–1680, 2012.
- [57] L. Duan, I. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.
- [58] Z. Guo and Z. J. Wang, "Cross-domain object recognition via input-output kernel analysis," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3108–3119, Aug 2013.
- [59] Y. Shi and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2012.
- [60] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," in *International Conference on Learning Representations*, 2013.
- [61] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *IEEE International Conference on Computer Vision*, 2013.
- [62] B. A. Olshausen and D. J. Fieldt, "Sparse coding with an overcomplete basis set: a strategy employed by v1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.
- [63] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1031–1044, 2010.
- [64] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, June 2010.
- [65] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa, "Domain adaptive dictionary learning," in *European Conference on Computer Vision*, vol. 7575, 2012, pp. 631–645.
- [66] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.
- [67] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *IEEE International Conference on Computer Vision*, 2013.
- [68] J. Zheng, R. Chellappa, and P. J. Phillips, "Sparse embedding-based domain adaptation for object recognition," in *IEEE International Conference on Computer Vision Workshop on Visual Domain Adaptation and Dataset Bias*, 2013.
- [69] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction," in *European conference on Computer vision*, 2012.
- [70] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation," in *International Conference on Machine Learning*, 2013.
- [71] B. Gong, F. Sha, and K. Grauman, "Overcoming dataset bias: An unsupervised domain adaptation approach," in *Neural Information Processing Systems Workshop on Large Scale Visual Recognition and Retrieval*, 2012.

- [72] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [73] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.
- [74] G. Hinton and R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504 – 507, 2006.
- [75] M. Chen, Z. Xu, and K. Q. Weinberger, “Marginalized Denoising Autoencoders for Domain Adaptation,” in *International Conference on Machine Learning*, 2012.
- [76] X. Glorot, A. Bordes, and Y. Bengio, “Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach,” in *International Conference on Machine Learning*, 2011.
- [77] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa, “Joint hierarchical domain adaptation and feature learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, submitted, 2013.
- [78] S. Chopra, S. Balakrishnan, and R. Gopalan, “DLID: Deep learning for domain adaptation by interpolating between domains,” in *ICML Workshop on Challenges in Representation Learning*, 2013.
- [79] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” in *International Conference on Machine Learning*, 2013.
- [80] L. Bo, X. Ren, and D. Fox, “Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms,” in *Neural Information Processing Systems*, 2011, pp. 2115–2123.
- [81] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, “Attribute and simile classifiers for face verification,” in *IEEE International Conference on Computer Vision*, October 2009.
- [82] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, “Describing objects by their attributes,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1778–1785.
- [83] Y. Wang and G. Mori, “A discriminative latent model of object classes and attributes,” in *European Conference on Computer Vision*, 2010, pp. 155–168.
- [84] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie, “Visual recognition with humans in the loop,” in *European Conference on Computer Vision*.
- [85] C. H. Lampert, H. Nickisch, and S. Harmeling, “Learning to detect unseen object classes by between-class attribute transfer,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 951–958.
- [86] A. Kovashka and K. Grauman, “Attribute adaptation for personalized image search,” in *IEEE International Conference on Computer Vision*, 2013.
- [87] T. Tommasi and B. Caputo, “Frustratingly easy nbnn domain adaptation,” in *IEEE International Conference on Computer Vision*, 2013.
- [88] S. S. F. V. Jain, “Adapting classification cascades to new domains,” in *IEEE International Conference on Computer Vision*, 2013.
- [89] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Computing Surveys*, pp. 399–458, Dec. 2003.
- [90] H. Ho and R. Gopalan, “Model-driven domain adaptation on product manifolds for unconstrained face recognition,” *International Journal of Computer Vision*, 2014.
- [91] R. Gross, I. Matthews, J. F. Cohn, T. Kanade, and S. Baker, “Multi-pie,” *Image Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

- [92] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *International Conference on Computer Vision*, 2011, pp. 543–550.
- [93] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Institute of Technology, Tech. Rep. 7694, 2007. [Online]. Available: <http://authors.library.caltech.edu/7694>
- [94] K. Gallivan, A. Srivastava, X. Liu, and P. Van Dooren, "Efficient algorithms for inferences on Grassmann manifolds," in *Statistical Signal Processing, 2003 IEEE Workshop on*, Sept 2003, pp. 315–318.
- [95] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," Technion - Computer Science Department, Tech. Rep. CS-2008-08, April 2008.
- [96] M. Elad, *Sparse and Redundant Representations: From theory to applications in Signal and Image processing*. Springer, 2010.
- [97] A. Yang, Z. Zhou, A. Balasubramanian, S. Sastry, and Y. Ma, "Fast l1-minimization algorithms for robust face recognition," *IEEE Transactions on Image Processing*, vol. 22, no. 8, pp. 3234–3246, Aug 2013.
- [98] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, Jan 2013.
- [99] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," UIUC Technical Report, Tech. Rep. UILU-ENG-09-2214, July 2009.
- [100] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," UIUC Technical Report, Tech. Rep. UILU-ENG-09-2215, Oct 2009.
- [101] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.